

REMARKS

This response is to the official action mailed in the above-referenced case on November 26, 2007, made final. Claims 1-11 are standing for examination.

Merit rejection under 35 U.S.C. 103(a)

Claims 1, and 3-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Batch as taught by "Not a Batch Language: A Control Language!", E.H. Bristol, published May 1995, in view of the implementation of the methodology of Object Oriented as taught by Object-Oriented Modelling and Simulation of Batch Plants (Wollhaf et al.) from November 1955, hereinafter Wollhaf.

Examiner's rejection

Batch teaches a software instance operating on a computer platform including a model framework for specifying a purpose-specific batch programs (Batch, page 2 - object oriented - by definition supports instances and Role of Graphics for framework support also see page 3) comprising: an extensible code library (Wollhaf, 00 Batch), an abstraction representing a batch program (Batch, page 5, Figure 3b and page 8 Figure 5); an abstraction representing a batch function of the program (Batch, page 12, operations); an abstraction representing operation of the function (as per above); an abstraction representing a data provider to the function (Operations above the operation often called a getter should be well known) ; and an abstraction presenting a context class of the function (Batch, as defined by the meta model provided by Inheritance, see page 2); characterized in that an instantiation process of the models is initiated with appropriate input data parameters input to each abstraction generating appropriate instances of batch functions and function operations wherein the generated instances are executable as part of a run sequence of the purpose-specific batch program (Batch, instantiating an object based on the class structure as taught on page 2 and the variables of the object as defined on page 7 and Wollhaf,

Chemical plants etc)). Batch teaches modeling batch programs using an Object Oriented Methodology. Wollhaf teaches the Object-Oriented modeling for a specific purpose of batchplants and the implementation of modeling (code library required to perform OO modeling) and the implementation of simulation. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Batch and Wollhaf, because Object technology provides a high degree of reuse with extensible code libraries.

Applicant's response

Applicant herein cancels claims 1-11 and presents new claims 12-23 for examination. Applicant's newly presented claims serve to recite the heart of applicant's invention which is the capability of automatic generation of executable data processing batch programs from their declarative specifications in the form of models. Applicant's invention views a data processing batch program to comprise of a 'fixed part' and a 'variable part'. The fixed part is common for all data processing batch programs and the variable part is specific to each batch program. Applicant's invention, as claimed, generates the variable part from its specifications and encapsulates the fixed part in the form of an extensible code library with placeholders where the variable parts can be plugged in. Our extensible code library also ensures that on failure a batch program will restart automatically, from a consistent state, with minimal loss of computation.

Applicant argues that the art of Batch pertains to the domain of industrial plants. Batch is utilizing a control language to specify how the static batch programs control an industrial plant. Applicant argues that the art of Batch cannot be applied in the domain of data processing batch programs. Wollhaf teaches a broad concept of the modeling and simulation of recipe driven multipurpose chemical or biochemical plants using an object oriented approach wherein a data model is presented which enables the description of all important aspects of batch plants. Applicant argues that

the art of Wollhaf cannot be applied in the domain of data processing batch programs.

Applicant's invention provides a high level mechanism for specifying application-specific variable parts of specific data processing batch programs from which their implementations can be automatically generated. Applicant's newly presented claim 12 recites an invariable part of all data processing batch programs presented as an extensible code library; at least one place-mark positioned at a point in the invariable part; at least one necessary variable part for insertion into the invariable part; and a model representing one declarative specification of the new data processing batch program; wherein the necessary variable parts are generated using a model and the generated variable parts are inserted into the invariable portion at the place-mark. Batch and Wollhaf, either singly, or in combination fail to show the unique advantages of automatically generating variable parts of a data processing batch program from its model and plugging the variable parts in the extensible code library at specified place-marks when operation of the data processing batch program requires it.

Applicant's newly presented system claim 18 includes the software used to generate the results as claimed in method claim 12. Therefore, claims 12 and 18 have been shown to be patentable over the art presented by the Examiner. Claims 13-17 and 19-23 are patentable on their own merits, or at least as depended from a patentable claim.

Applicant respectfully requests reconsideration and the case be passed quickly to issue. If there are any time extensions required for response in addition to any extension petitioned and paid with this response, such petition is requested, and if there are any fees due over any fees paid with this response, authorization is given to deduct the fees from deposit account 50-0534.

Respectfully submitted,
Vinay Vasant Kulkarni et al.

By *Donald R. Boys*
Donald R. Boys
Reg. No. 35,074

Central Coast Patent Agency, Inc.
3 Hangar Way, Suite D
Watsonville, CA 95076
831-768-1755